Visual Marker Guided Point Cloud Registration in a Large Multi-Sensor Industrial Robot Cell

Erind Ujkani*, Joacim Dybedal*, Atle Aalerud*, Knut Berg Kaldestad* and Geir Hovland*
*University of Agder, Norway
Department of Engineering Sciences
Mechatronics Group
Email: {erindu12, joacim.dybedal, atle.aalerud, knut.kaldestad, geir.hovland} [at] uia.no

***Abstract*—This paper presents a benchmark and accuracy analysis of 3D sensor calibration in a large industrial robot cell. The sensors used were the Kinect v2 which contains both an RGB and an IR camera measuring depth based on the time-of-flight principle. The approach taken was based on a novel procedure combining Aruco visual markers, methods using region of interest and iterative closest point. The calibration of sensors is performed pairwise, exploiting the fact that time-of-flight sensors can have some overlap in the generated point cloud data. For a volume measuring 10m x 14m x 5m a typical accuracy of the generated point cloud data of 5-10cm was achieved using six sensor nodes.**

I. INTRODUCTION

With an increasing quality and availability of 3D sensors, there is a growing interest to use such sensors in industrial applications. The use of these sensors could open up new and more flexible applications, for example human-machine interaction, as compared to traditional applications with safety fences and restricted access for humans.

One of the first steps required when setting up an industrial cell instrumented with 3D sensors is calibration, ie. to match the internal coordinate systems for each sensor such that the point cloud from each sensor can simply be added together to form a global point cloud for the entire industrial cell.

In [1] a system for calibrating 28 Asus Xtion 3D sensors based on structured light and 6 Hokuyo 2D lasers was presented. The method used tracking of pedestrian heads for sensor calibration. In [1] it is stated: *Although the 3D sensors suffer from a low degree of overlap, the addition of 2D sensors, which have a much wider coverage area, improves the number of shared observations, resulting in a higher level of calibration accuracy.* 3D sensors based on structured light can not have a high degree of overlap because of interference of the emitted structured light between the sensor nodes. In [2] a solution was presented where vibration motors were attached to each sensor to mitigate the problem. However, this solution reduces the accuracy of the sensors as well as increases complexity. In our work 3D sensors based on structured light were avoided, and sensors based on the time-of-flight (ToF) principle were used instead, which allows for overlap between the different 3D sensors. Additionally, it is stated in [3] that outdoor applicability for structured light based depth sensors is usually hard to achieve.

The sensors used in our work were six Kinect V2 which contain an RGB camera in addition to a time-of-flight based depth sensor. To achieve minimum distortion with the Kinect V2 the calibration of the intrinsic parameters and distortion coefficients of the IR camera is important. In [1] it is stated that a large number of visual markers will be necessary to achieve good calibration when a large number of sensors is used. This is true if the entire sensor system needs to be calibrated in one operation. In our work this drawback is overcome by using movable visual markers to calibrate one pair of 3D sensors at a time, exploiting the overlap between the sensor nodes. Our considered system will introduce a novel combination of visual markers and 3D point clouds to estimate the relative pose between cameras. Additionally, an algorithm was created to select the pairwise calibration order when using the point clouds.

Visual markers are widely used for augmented reality (AR) applications. These provide camera-marker relative pose estimations, that in our case were used to estimate pose for several cameras in relation to each other. A visual marker from [4] called Aruco markers and the OpenCV library from [5], were used to perform this operation, which is also known as extrinsic calibration. A two-step fine tuning of the Aruco pose estimation is done using the ICP algorithm from the Point Cloud Library (PCL) [6].

II. PROBLEM FORMULATION AND MOTIVATION

In an industrial robot cell consisting of depth cameras, it was desired to determine the pose of the cameras using visual markers and depth data. It was defined that pose estimations that do not differ markedly from a manual estimation, and point cloud merging that is able to reconstruct an industrial workspace within certain criteria, are characteristics of a well-performing system.

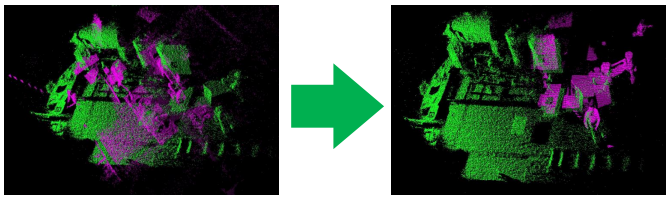

Fig. 1. Figure showing the registration process on the workspace.

978-1-5386-4643-4/18/$31.00 ©2018 IEEE

III. METHODS

A. Calibration

Camera calibration is essential for improving the results and performance of the considered system. The calibration is performed using the provided tools in [7]. This material is well-known in image processing, but is included here for completeness. The process is divided into three sub-calibrations:

RGB camera: Here *distortion* and the *intrinsics* need to be addressed by parameter estimation. As the name itself implies distortion of the camera picture describes the warping being present. The field of camera distortion is a study of its own, but in our case it is modeled as the sum of a *tangential* and *radial* distortion [8]. A distorted point on the image plane (x_d, y_d) can be modeled to an undistorted point (x_u, y_u) by calculating the radial coefficients k_1, k_2, k_3 and tangential coefficients p_1, p_2. Radial distortion is modeled as follows:

$$\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} x_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{bmatrix} \quad (1)$$

and tangential distortion is modeled:

$$\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} x_d + (2p_1 xy + p_2(r^2 + 2x^2)) \\ y_d + (p_1(r^2 + 2y^2) + 2p_2 xy) \end{bmatrix} \quad (2)$$

For an image center point (x_c, y_c), r is simply the radius:

$$r = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2} \quad (3)$$

Depth camera: Calibration of the depth estimation consists of estimating the same parameters as the RGB camera. The only difference is that the camera calibrated is an IR camera, leading to different intrinsics and distortion coefficients being estimated.

RGB + Depth camera: This calibration is important for combining the RGB and depth camera by relating color and depth. By utilizing the previous calibrations of the RGB camera and depth camera the extrinsics, which relate the position and orientation between these two cameras, are calculated.

B. Aruco transformation estimation

As shown in the flowchart in Fig. 2 the first step in the registration process is using the RGB camera to perform the Aruco transformation estimation. Here, a transformation is estimated from the camera frame to the Aruco frame using the provided API [4]. Let H_A^1 denote the transformation from the first camera to the Aruco frame and H_A^2 the transformation from the second camera to the same Aruco frame. It then follows that the transformation from the first camera to the second camera is:

$$H_2^1 = H_A^1 \cdot (H_A^2)^{-1} \quad (4)$$

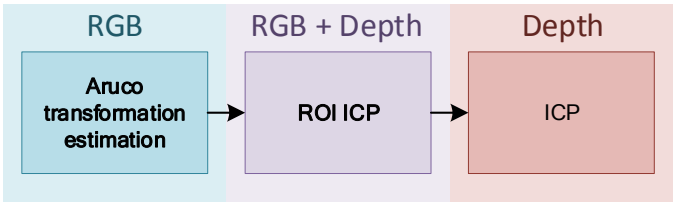

Fig. 2. Flowchart showing the Aruco guided ICP registration

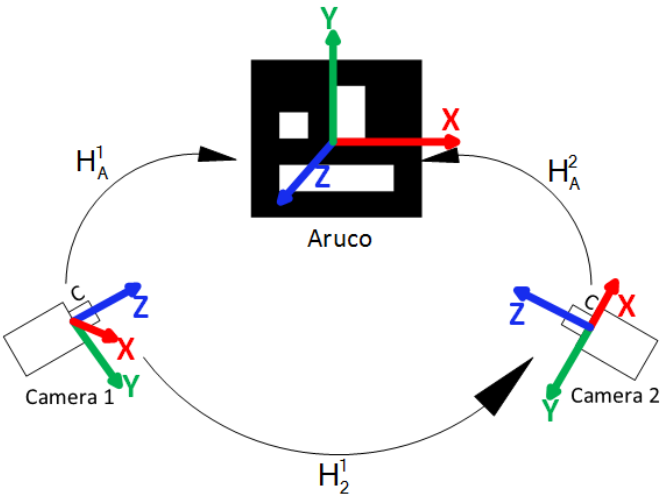

Fig. 3. Figure showing the Aruco transformations and frames in the system.

C. Region of Interest (ROI) based Iterative Closest Point (ICP)

The next step in the presented registration technique considers further use of the Aruco markers to provide a region of interest on the point cloud. Here, the 4 marker corners provided from the Aruco transformation estimation on the RGB camera was utilized to perform this procedure. By using [7] a rectified depth map which is upscaled to the same resolution as the RGB camera can be acquired. This means that every coordinate on the depth map has an associated color and thus a corresponding depth data region of interest can be extracted. The depth map is then converted to 3D points using the perspective projective equations and casted into a PCL point cloud. For our considered system three different Aruco markers were used to segment three clusters in a point cloud ROI. This process also utilizes the transformation provided by the Aruco RGB transformation estimation as an initial estimate. The final step is to perform an alignment by using the ICP algorithm from [6] between the ROI segmented point clouds. As a result an adjustment transformation matrix, H_{ICP1}, is calculated and eq. (4) is then updated to:

$$H_2^1 = H_A^1 \cdot (H_A^2)^{-1} \cdot H_{ICP1} \quad (5)$$

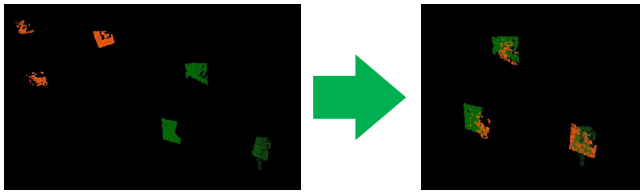

Fig. 4. ROI registration defined by the Aruco marker corners.

D. Complete cloud ICP

The two previous steps guide the cloud to a more optimal search space for the final step in Fig. 2. Since the previous step can not extract all of the overlapping 3D points, a final registration process can be performed by using the complete clouds. In order to increase the number of correspondences, the clouds from each corresponding sensor were accumulated over two iterations. Here the cloud to be registered is initially oriented and positioned by the transformation from eq. (5). The final transformation matrix, which uses the complete cloud ICP transformation matrix H_{ICP2}, is then updated to:

$$H_2^1 = H_A^1 \cdot (H_A^2)^{-1} \cdot H_{ICP1} \cdot H_{ICP2} \tag{6}$$

E. Multi-sensor registration procedure

Since the presented registration method is a matching process between two clouds, registration of multiple cameras simultaneously is not possible. However, two algorithms were created in order to match the clouds to the reference cloud as seen in Fig. 6. The first algorithm calibrates each cloud to the reference cloud by use of the steps shown in Fig. 2. Since little overlap is available between the cameras, a second algorithm is studied (Alg. 1). The underlying difference of this algorithm is that it iteratively chooses the best fitting cloud in relation to the reference cloud based on the ICP score metric and merges it to the reference cloud. The merged cloud is removed from the calibration clouds array and the process continues with new calculated ICP scores until all clouds are calibrated. This is done in order to increase the number of correspondent point pairs when performing the final fine registration. Additionally, it is assumed that sensor 4 is pose estimated with respect to a world frame beforehand. This is done to prevent accumulation of error when comparing the manual calibrations from [9] with the calibrations using the presented algorithms.

IV. EXPERIMENTAL SETUP

The considered system consists of a large industrial test facility monitored by six Kinect v2 RGBD cameras placed and oriented as shown in Fig. 5. Each camera is connected to a Jetson TX2 developer board to perform preprocessing of the 3D data, which is then sent to a central computer for further processing. More information about the connection setup and use of the Robot Operating System (ROS) as a middleware can be found in [10]. The cameras were placed in heights of 4.8 to 5.1m in an industrial test facility with an area of 10m x 14.5m. Three Aruco markers of size 48x48cm were used, drawn on plates of size 1.2m x 0.6m. More information about the test facility can be found in [9].

Algorithm 1 Pseudo code for the calibrate and merge algorithm.

```
// reference cloud RC = 4
// calibration clouds CC = (1, 2, 3, 5, 6)
// final ICP results array = ICP_res
while CC.size() ≠ 0 do
    empty ICP_res
    for every element in CC do
        registrate with respect to RC
        store final ICP in ICP_res
    end for
    select lowest ICP score in ICP_res
    merge corresponding cloud to RC
    remove cloud added to RC from CC
end while
```

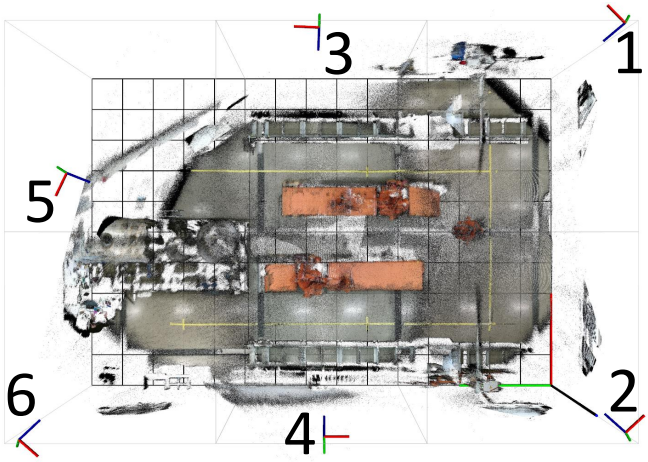

Fig. 5. RGBD overview of the workspace covered by the Kinect V2 sensors.

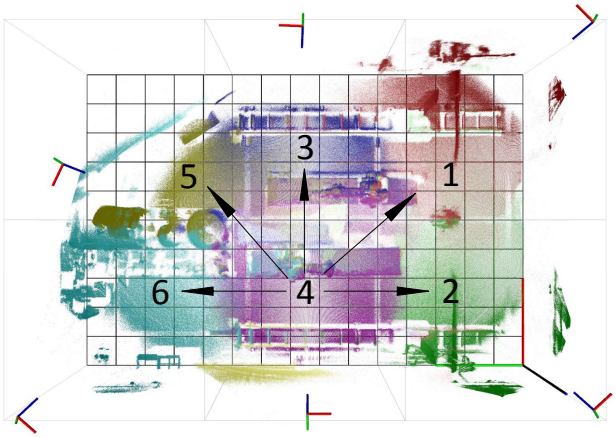

Fig. 6. Color clouds with the corresponding camera numeration. Note that kinect 4 is used as a reference in the registration process.

Fig. 7. Picture of the industrial test facility.

V. EXPERIMENTAL RESULTS

Figures 10 and 11 present Aruco accuracy results by using a 28x28cm Aruco marker with ground truth provided by the Leica Absolute Tracker AT960 and RGB images provided by the Kinect V2. Both Fig. 10 and Fig. 11 are results of depth analysis with no variations of Aruco orientation. The Aruco plate was also kept coplanar with respect to the camera. Extrinsic calibration results are presented by two different tests. The first test compares the extrinsic parameters between a manual calibration (MC) from [9] and a calibration using the presented algorithms. Table I shows the pose difference of the pairwise calibration (PC) of clouds 4-1, 4-2, 4-3, 4-5 and 4-6 according to Figs. 2 and 6, whilst Table II shows the standard deviations of these pose estimations.

Table III shows the calibration difference using Alg.1. The algorithm gave a calibration order of 6, 3, 5, 2, 1 based on the ICP fitness scores. In Table IV the Kinect 1 is guided to better results by placing a plate of size 1.2x0.6m between sensor 1 and 2 for an increase in correspondent point pairs. Taking this improvement into account, a standard deviation of this pose estimation is shown in Table V. Further, a second test measures the correspondence between physical points and points from the point cloud. Here, an Euclidean distance on the xy-plane as shown in Fig. 8, is measured between two points using retro-reflective tape as a visual marker. The distances were read manually by inspection of the point cloud and compared with physical measurements provided by the Laser Distance Meter (LDM) Leica Disto D4a BT. Tables VI and VII show the errors calculated by using the pose estimations used in Tables III and IV respectively.

TABLE I
POSE ESTIMATION DIFFERENCE BETWEEN MC AND PC.

Sensor	Difference [cm and deg]					
	x	y	z	R_z	R_y	R_x
Kinect 1	31.8	-13.6	-2.9	0.735	0.957	0.770
Kinect 2	-18.5	15.4	11.3	-2.362	-0.122	-0.758
Kinect 3	13.9	-38.0	-9.6	-3.769	0.350	1.127
Kinect 5	-9.1	7.8	-11.8	1.708	-0.638	1.052
Kinect 6	1.0	4.1	-7.1	-0.662	0.609	0.883

TABLE II
POSE ESTIMATION STANDARD DEVIATION OF THE PC.

Sensor	Standard deviation [cm and deg]					
	x	y	z	R_z	R_y	R_x
Kinect 1	13.54	9.10	7.07	0.738	0.156	0.584
Kinect 2	4.98	3.43	5.46	0.525	0.312	0.440
Kinect 3	3.67	16.88	2.58	1.766	0.097	0.211
Kinect 5	7.35	3.90	1.16	0.685	0.065	0.136
Kinect 6	2.23	2.73	4.13	0.237	0.222	0.375

TABLE III
POSE ESTIMATION DIFFERENCE BETWEEN MC AND ALG.1.

Sensor	Difference [cm and deg]					
	x	y	z	R_z	R_y	R_x
Kinect 1	28.8	4.0	-10.0	1.810	0.113	-0.414
Kinect 2	-24.3	6.8	2.9	-1.894	-0.950	-0.441
Kinect 3	15.1	-8.1	-8.0	-0.681	0.903	0.991
Kinect 5	-6.6	1.4	-12.6	-0.152	-0.105	0.885
Kinect 6	1.5	-7.3	-6.4	-0.005	-0.349	0.334

TABLE IV
UPDATED POSE ESTIMATION DIFFERENCE OF KINECT 1 USING ALG.1.

Sensor	Difference [cm and deg]					
	x	y	z	R_z	R_y	R_x
Kinect 1	-11.90	-10.84	5.41	-1.475	0.804	0.164

TABLE V
STANDARD DEVIATION OF POSE ESTIMATION USING ALG.1.

Sensor	Standard deviation [cm and deg]					
	x	y	z	R_z	R_y	R_x
Kinect 1	10.66	5.64	2.78	1.102	0.273	0.224
Kinect 2	2.00	2.85	2.54	0.267	0.271	0.304
Kinect 3	2.41	6.82	3.58	0.753	0.231	0.325
Kinect 5	1.74	1.02	0.99	0.125	0.103	0.104
Kinect 6	2.96	1.97	2.48	0.194	0.182	0.246

TABLE VI
TABLE SHOWING THE RESULTING ERRORS BETWEEN PHYSICALLY MEASURED DISTANCES AND POINT CLOUD DISTANCES (SEE FIG. 8).

Measurement	Actual[m]	Measured[m]	Error[m]
Between yellow crosses	6.722	6.469	0.253
Between red crosses	6.686	6.604	0.082
Between violet crosses	1.484	1.525	-0.041
Between blue crosses	1.548	1.531	0.017

TABLE VII
TABLE SHOWING IMPROVED RESULT OF MEASUREMENT "BETWEEN YELLOW CROSSES" (AS ILLUSTRATED IN FIG. 8).

Measurement	Actual[m]	Measured[m]	Error[m]
Between yellow crosses	6.722	6.785	-0.063

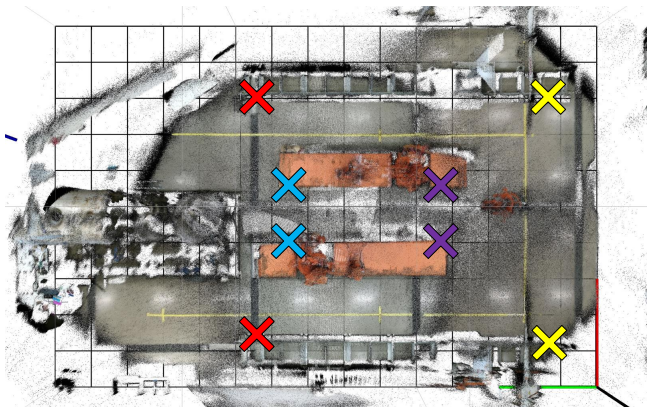

Fig. 8. Measurement points on the physical workspace. Each color represents the endpoints of a distance to be measured.

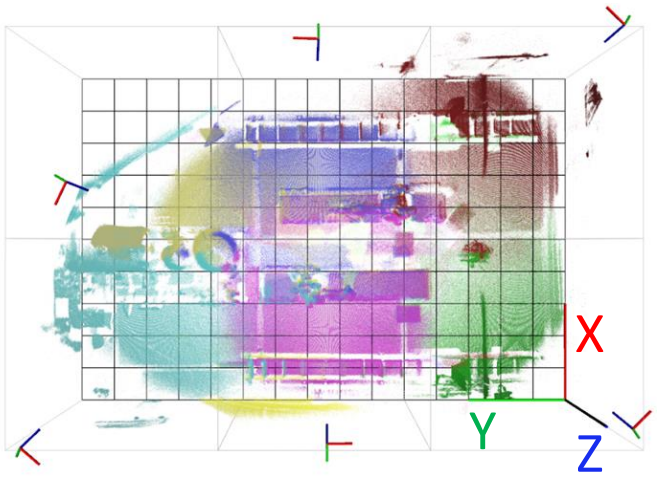

Fig. 9. Top view showing the calibration result and the global frame.

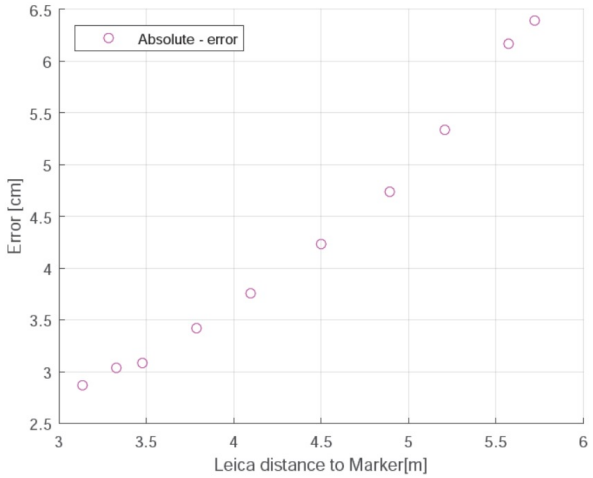

Fig. 10. Absolute distance error of a 28x28 cm Aruco marker with respect to a varying measurement distance.

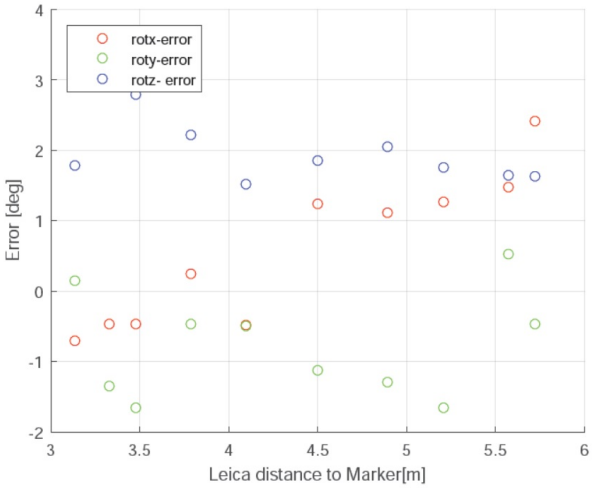

Fig. 11. Angle error of a 28x28 cm Aruco marker with respect to a varying measurement distance. The reference coordinate system is the local frame of the Leica AT960 Absolute Tracker.

VI. DISCUSSION & CONCLUSIONS

In this paper the results of a marker based registration algorithm have been presented. No timing results have been presented since the system does not have any strict calculation time requirements. This means that the algorithm runs offline and calibrates the sensor poses once as an initial step. It is assumed that the cameras do not experience any external influences such as vibrations. If continuous registration is needed, such as in the case of minor pose changes from external vibrations, a GPU based ICP could be introduced after the initial pose is estimated using the presented algorithm.

The lighting condition is also crucial for the depth acquisition. Since the ToF sensor uses IR illumination on the scene, noise from IR light sources such as sunlight causes loss of weaker illuminated points. In our case the noise in form of IR light wa minimal since the test was performed in an indoor LED illuminated test facility. Another well known problem for depth sensors such as ToF sensors is the weak to non-existing depth estimation of darker objects. These problems can be dealt with but not completely solved by using ToF sensors with stronger IR illumination of the physical scene. This will result in a higher signal to noise ratio (SNR) when dealing with disturbances such as sunlight. A more complete documentation analyzing other ToF error factors such as warm-up time, internal scattering and wiggling can be found in [11]. Since the ToF sensors have a range dependent error as mentioned in [11] and [12], an object registered by one camera is not identically measured by another camera. This will result in difficulties for the ICP algorithm to find optimal correspondences. The generation of more correct correspondent point pairs between the cameras is also dependent on the reflectivity of the material. This was observed when utilizing retro-reflective tape, which has ideal reflectivity. Here, the 3D point generation was studied to be more precise and stable than e.g. matte colored surfaces. As mentioned earlier the clouds to be used on the final stage of

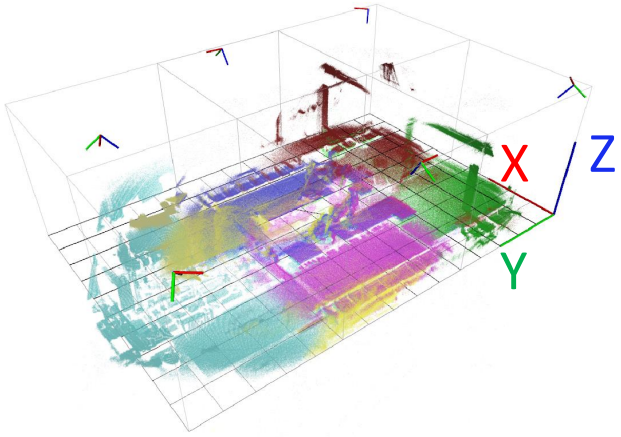

Fig. 12. Side view of the calibrated workspace.

the presented algorithm were accumulated over two iterations. An increase in the number of accumulated points is possible, but will of course influence the calculation time.

From the accuracy analysis results as shown in Figs. 10 and 11 it is seen that for an Aruco marker of size 28x28 cm the absolute distance error is around 6 cm at a distance of around 5.5 m. It was therefore desired to increase the dimensions of the Aruco marker in order to reduce the error. The error was observed to be non-linear and was strongly dependent on the intrinsic calibration and the Aruco orientation.

Other markers such as chessboards were considered but experienced difficulties. The strength of the chessboard is also its biggest weakness. The chessboard pattern consists of many smaller squares, where points are generated at each transition of these squares. This gives significantly more points for pose estimation than an Aruco marker, but at an expense. When the chessboard is slightly angled and/or at a large distance the detection of the pattern becomes more difficult compared to an Aruco marker of similar size. Nonetheless, the detection of the corner points of the Aruco, which are used to pose estimate, are strongly dependent on the size of the padding area. This area is important for the sub-pixel corner estimation process implemented from [5].

Tables II and V show the limitations and robustness of the considered system by means of standard deviation (STD) of the pose estimation. Considering the error rates and uncertainty of the used sensor a STD of around 4 cm in x, y and z direction is expected. A STD higher than 4 cm in the x, y and z direction is correlated to the lack of features and overlap in these directions. By using Table II as a reference, an object was placed between sensor 1 and 2 to increase the number of correspondences in x-direction. This process decreased the STD of the pose estimation of kinect 1 when using algorithm 1. It is thus shown that the robustness is strongly dependent on the number of correspondences. The results showing the difference between a manual point cloud calibration and the proposed solution were used only as a comparative analysis and have no direct implication of physical correctness. The direct comparison between the calibrated clouds using the proposed algorithm and physical points show an accuracy of 5-10 cm. Overall the considered system is applicable in implementations such as [13], where safety measures such as a relatively large Euclidean collision distance and robot point cloud padding are used.

ACKNOWLEDGMENT

The research presented in this paper has received funding from the Norwegian Research Council, SFI Offshore Mechatronics, project number 237896.

REFERENCES

[1] D. F. Glas, D. Brscic, T. Miyashita, and N. Hagita, "SNAPCAT-3D: Calibrating Networks of 3D Range Sensors for Pedestrian Tracking," *IEEE International Conference on Robotics and Automation*, 2015.

[2] D. A. Butler, S. Izadi, O. Hilliges, D. Molyneaux, S. Hodges, and D. Kim, "Shake'n'sense," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, p. 1933, 2012.

[3] H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect range sensing: Structured-light versus Time-of-Flight Kinect," *Computer Vision and Image Understanding*, vol. 139, pp. 1–20, 10 2015.

[4] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[5] "Open Source Computer Vision Library 3.3," OpenCV team, https://opencv.org/, 2017.

[6] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, (Shanghai, China), May 9-13 2011.

[7] T. Wiedemeyer, "iai_kinect2," Institute for Artificial Intelligence, University Bremen, https://github.com/code-iai/iai_kinect2, 2015.

[8] F. Bukhari and M. Dailey, "Automatic Radial Distortion Estimation from a Single Image," *Journal of Mathematical Imaging and Vision*, 2013.

[9] A. Aalerud, J. Dybedal, E. Ujkani, K. B. Kaldestad, and G. Hovland, "Industrial Environment Mapping using Distributed Static 3D Sensor Nodes," MESA, 2018.

[10] J. Dybedal, A. Aalerud, and G. Hovland, "Embedded GPU-Based Fusion of 3D Sensor Data," Submitted to IROS, 2018.

[11] P. Fürsattel, S. Placht, M. Balda, C. Schaller, H. Hofmann, A. Maier, and C. Riess, "A Comparative Error Analysis of Current Time-of-Flight Sensors," *IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING*, vol. 2, no. 1, 2016.

[12] J. Illade-Quinteiro, V. M. Brea, P. López, D. Cabello, and G. Doménech-Asensi, "Distance measurement error in time-of-flight sensors due to shot noise.," *Sensors (Basel, Switzerland)*, vol. 15, pp. 4624–42, 2 2015.

[13] E. Ujkani, P. S. Eppeland, A. Aalerud, and G. Hovland, "Real-time human collision detection for industrial robot cells," in *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 488–493, IEEE, 11 2017.